FIG. 1

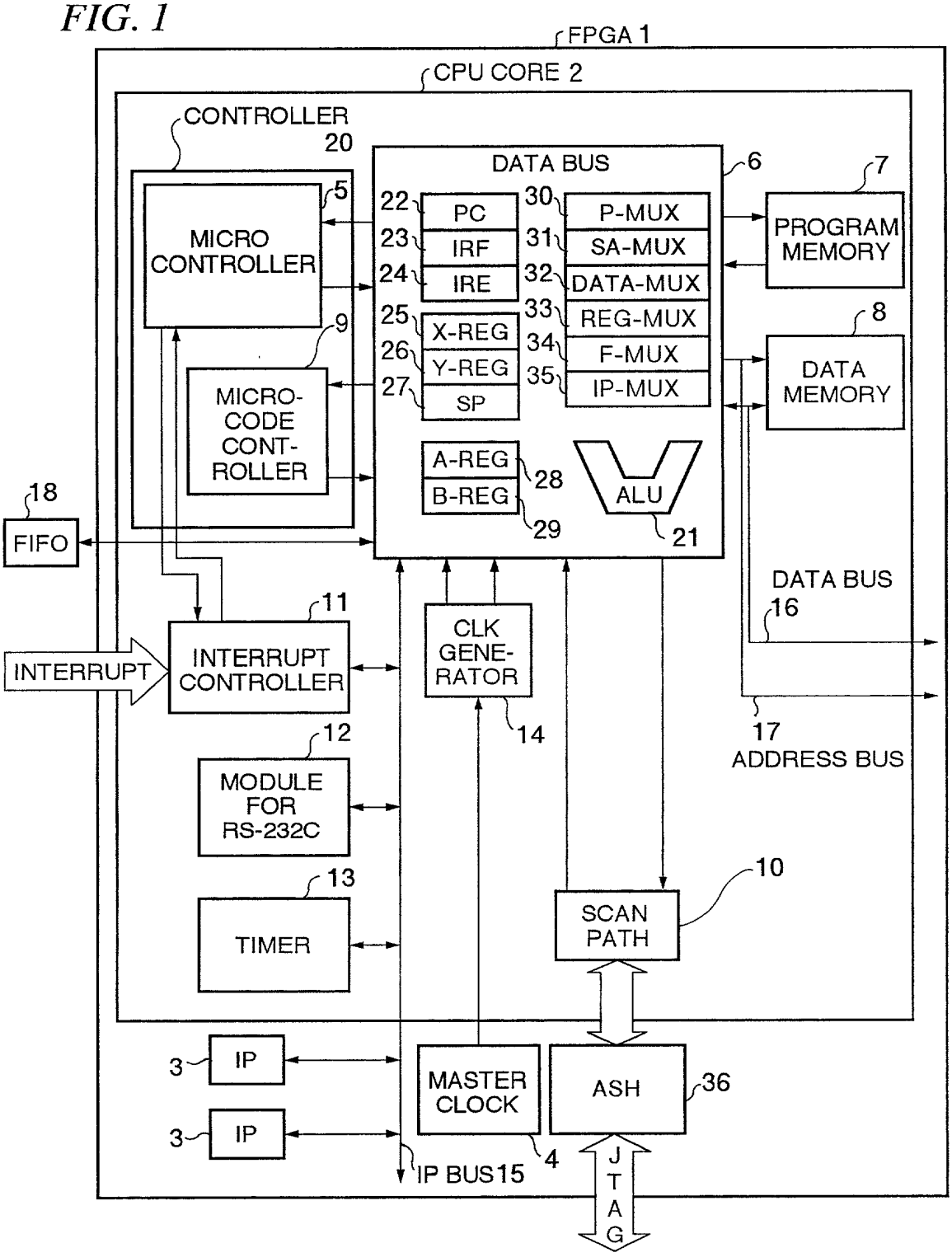

FPGA 1
CPU CORE 2
CONTROLLER
20
MICRO CONTROLLER
5
MICRO-CODE CONT-ROLLER
9
DATA BUS
6
22 PC
23 IRF
24 IRE
25 X-REG
26 Y-REG
27 SP
30 P-MUX
31 SA-MUX
32 DATA-MUX
33 REG-MUX
34 F-MUX
35 IP-MUX
A-REG
B-REG
28
29
ALU
21
7
PROGRAM MEMORY
8
DATA MEMORY
18
FIFO
INTERRUPT
11
INTERRUPT CONTROLLER
CLK GENE-RATOR
14
DATA BUS
16
17
ADDRESS BUS
12
MODULE FOR RS-232C
13
TIMER
10
SCAN PATH
3
IP
3
IP
MASTER CLOCK
4
ASH
36
IP BUS15
JTAG

FIG.2

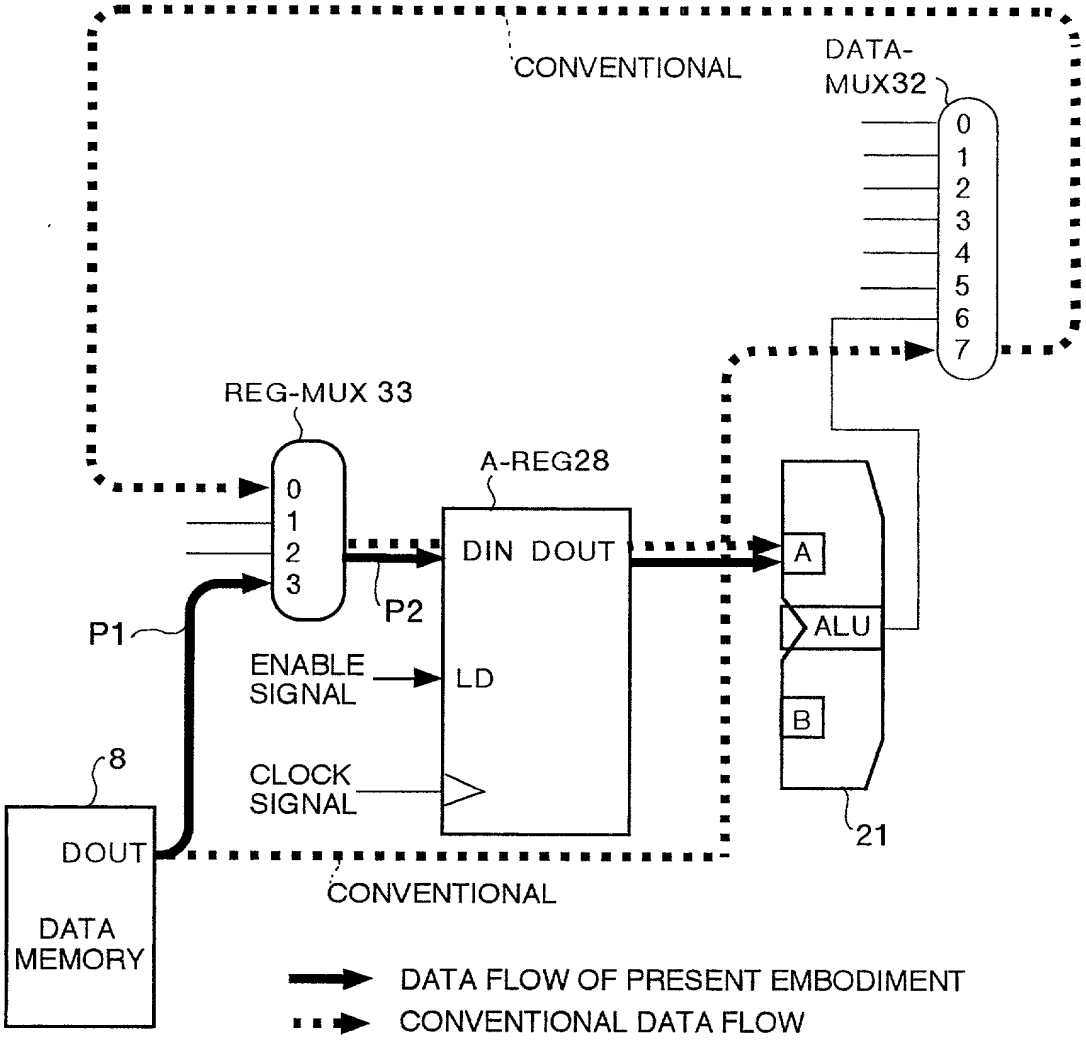
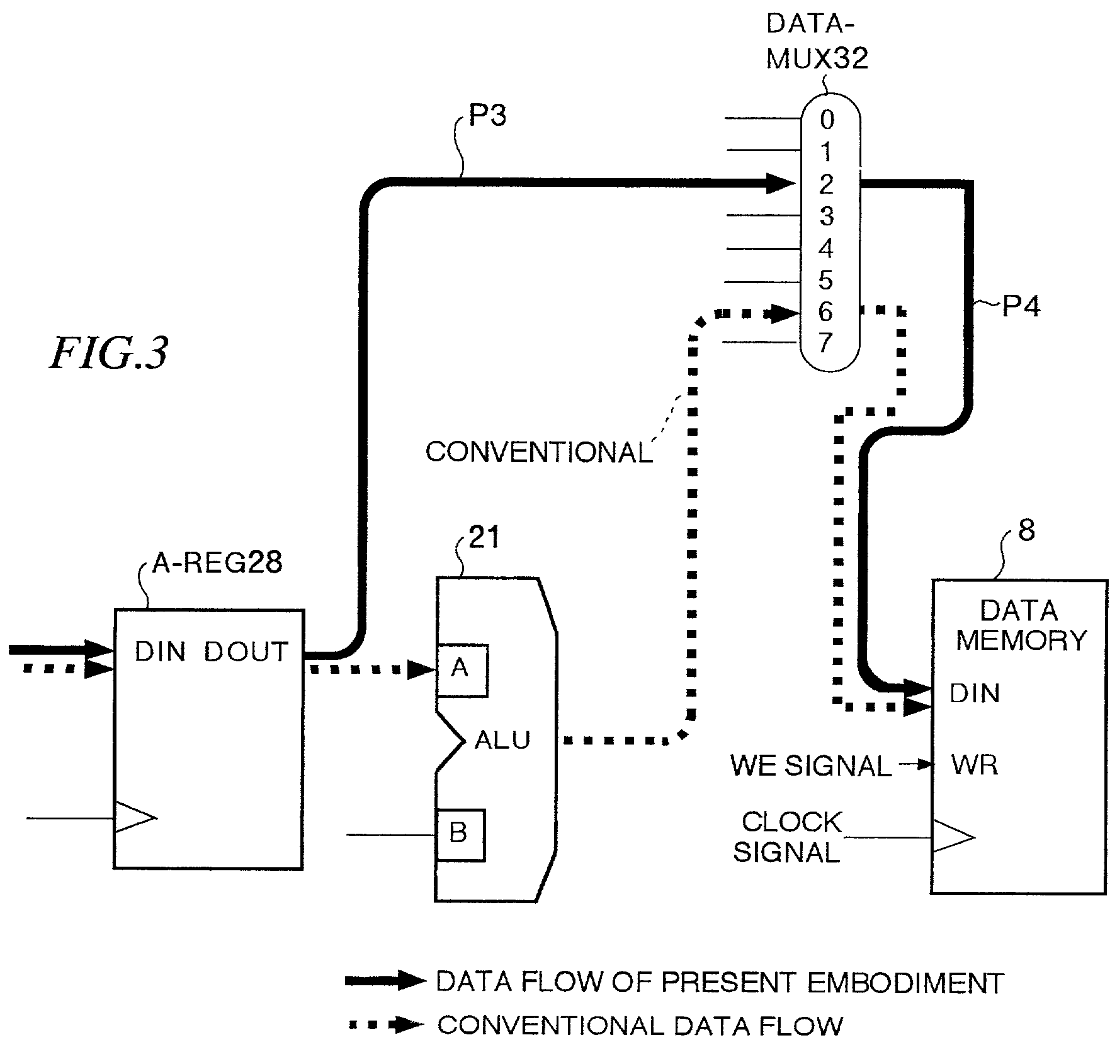

CONVENTIONAL
DATA-
MUX32
0
1
2
3
4
5
6
7
REG-MUX 33
0
1
2
3
A-REG28
DIN DOUT
P1
P2
ENABLE
SIGNAL
LD
CLOCK
SIGNAL
A
ALU
B
21
8
DOUT
DATA
MEMORY
CONVENTIONAL
DATA FLOW OF PRESENT EMBODIMENT
CONVENTIONAL DATA FLOW

FIG.3


DATA-
MUX32
0
1
2
3
4
5
6
7
P3
P4
CONVENTIONAL
A-REG28
DIN DOUT
21
A
ALU
B
8
DATA
MEMORY
DIN
WE SIGNAL
WR
CLOCK
SIGNAL
DATA FLOW OF PRESENT EMBODIMENT
CONVENTIONAL DATA FLOW

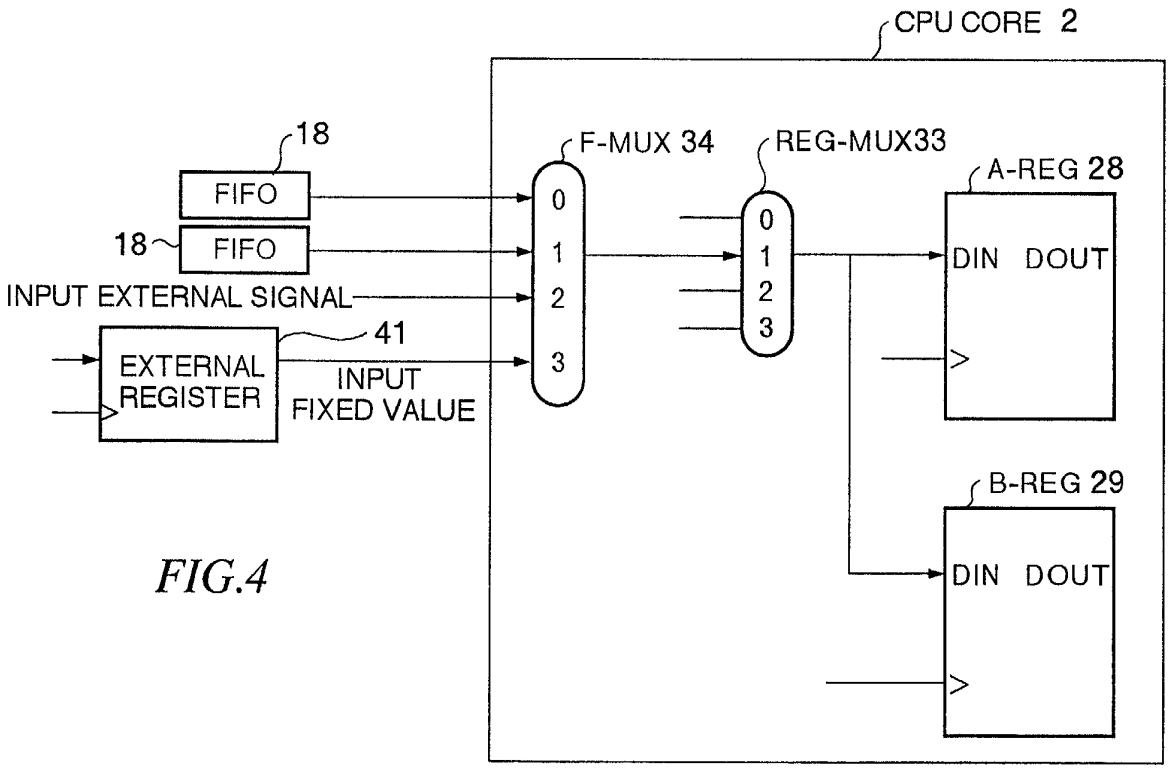

CPU CORE 2
18
FIFO
18
FIFO
INPUT EXTERNAL SIGNAL
EXTERNAL REGISTER
41
INPUT FIXED VALUE
F-MUX 34
0
1
2
3
REG-MUX33
0
1
2
3
A-REG 28
DIN DOUT
B-REG 29
DIN DOUT

FIG.4

FIG.5

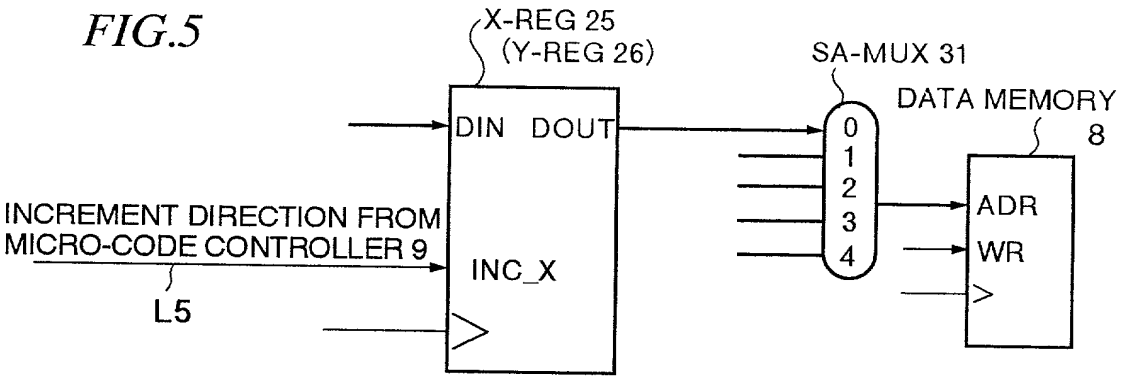

X-REG 25
(Y-REG 26)
SA-MUX 31
DATA MEMORY
8
DIN
DOUT
0
1
2
3
4
ADR
WR
INCREMENT DIRECTION FROM
MICRO-CODE CONTROLLER 9
INC_X
L5

FIG.6 PRIOR ART

EXAMPLE OF CONTROL IN SINGLE CLOCK (IN LDA & STA CONTINUOUS COMMAND)

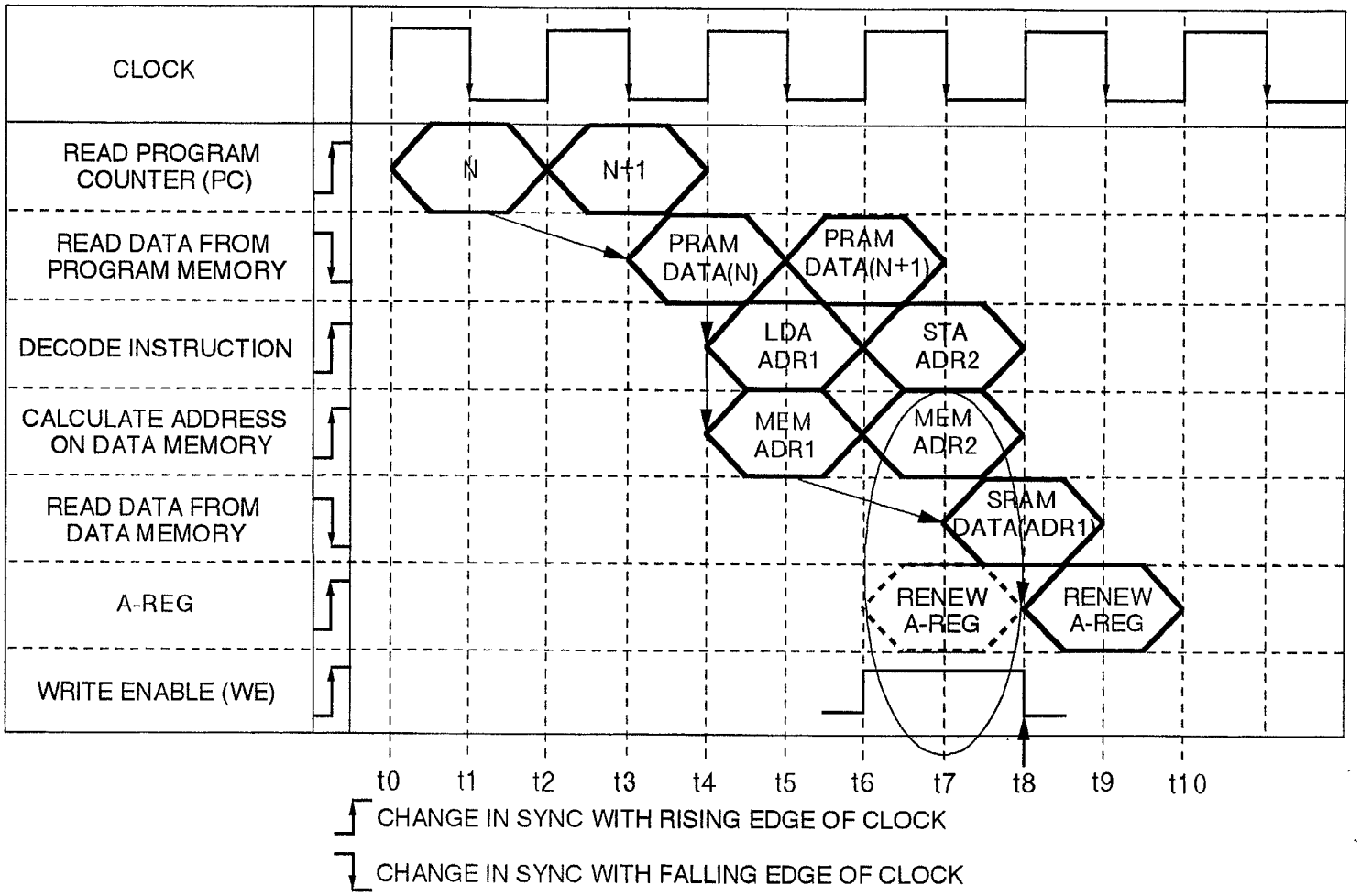
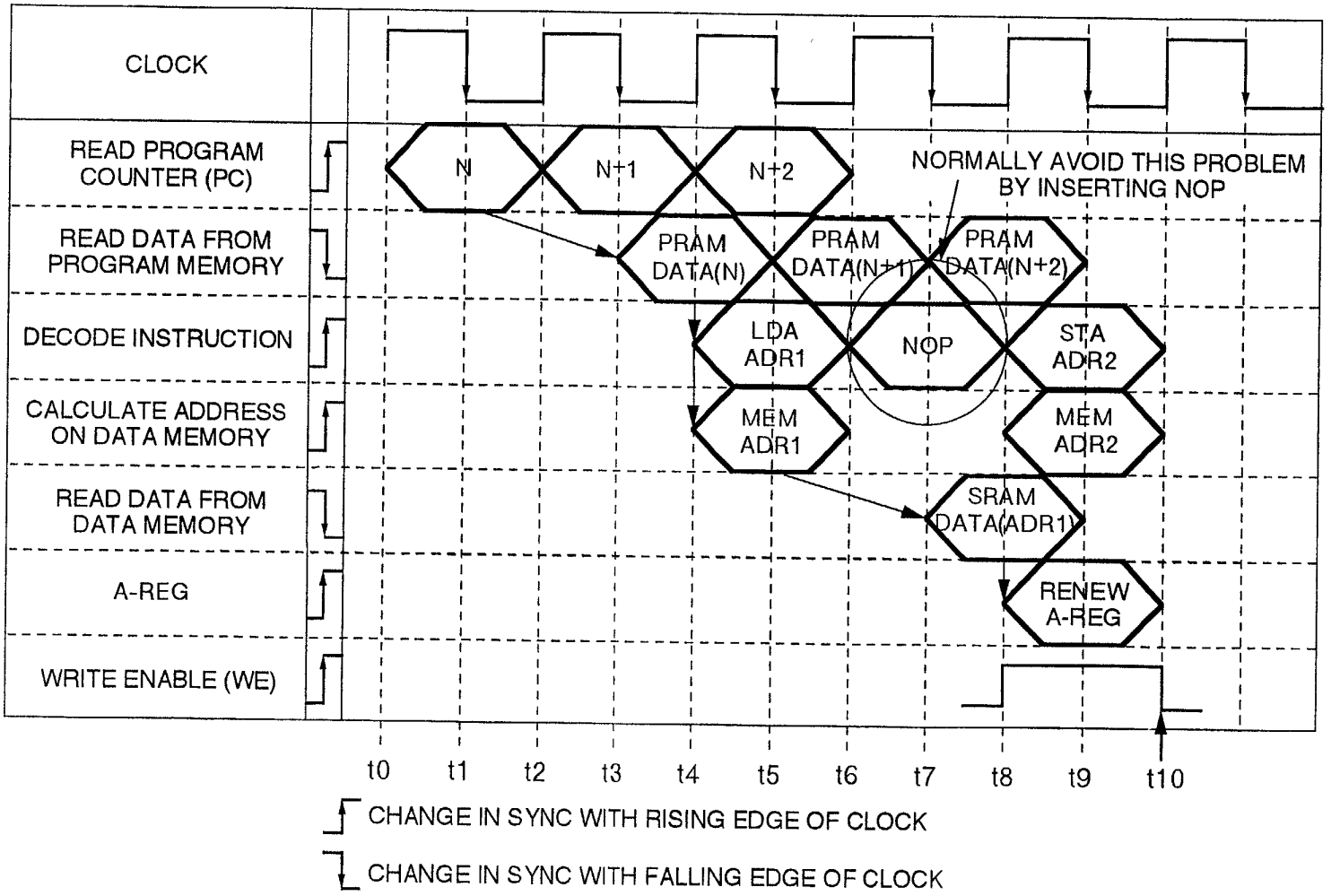


FIG.7

EXAMPLE OF SOLUTION BY INSERTING NOP



CONTROL METHOD OF CPU CORE IN PRESENT EMBOMDIMENT

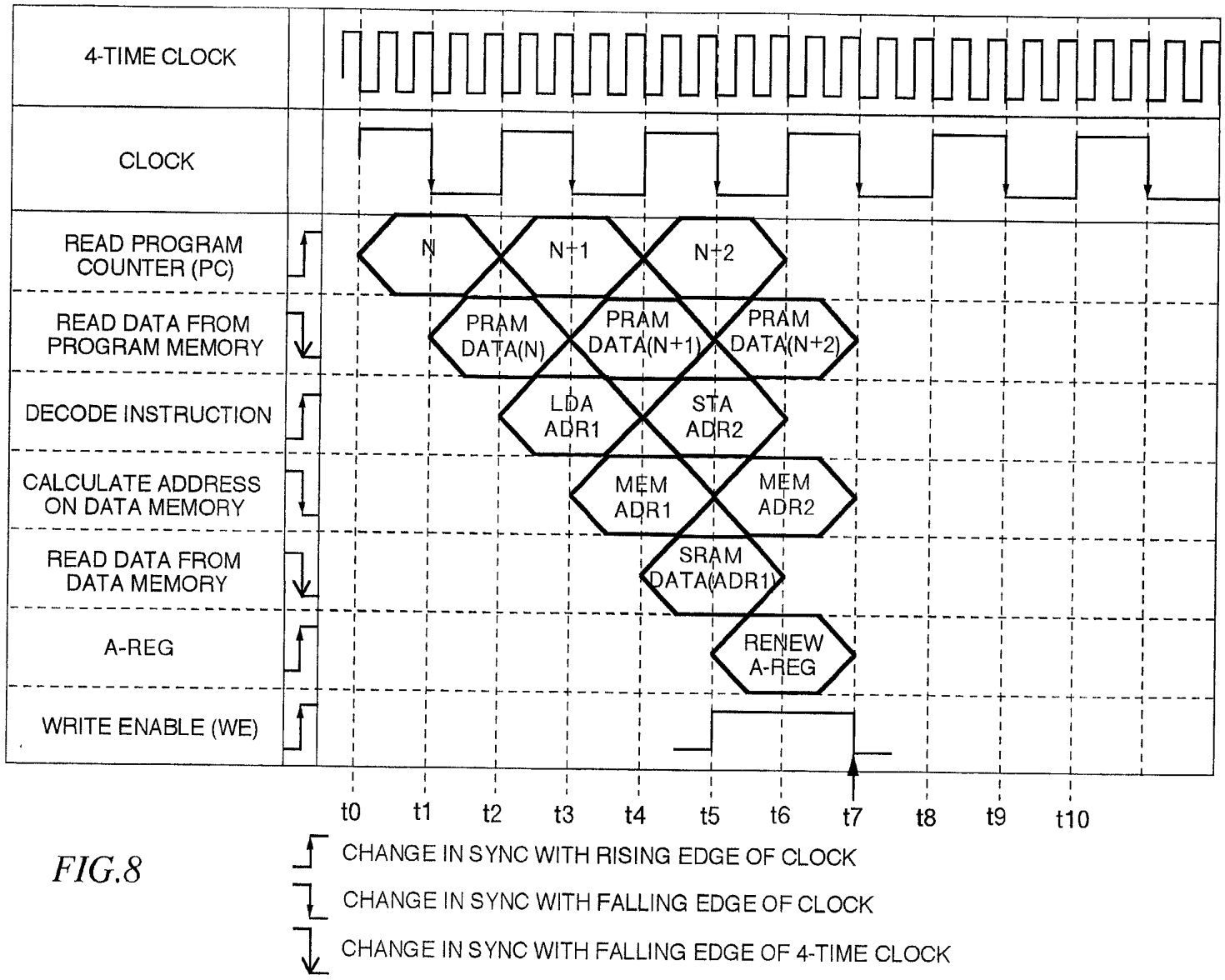


FIG.8

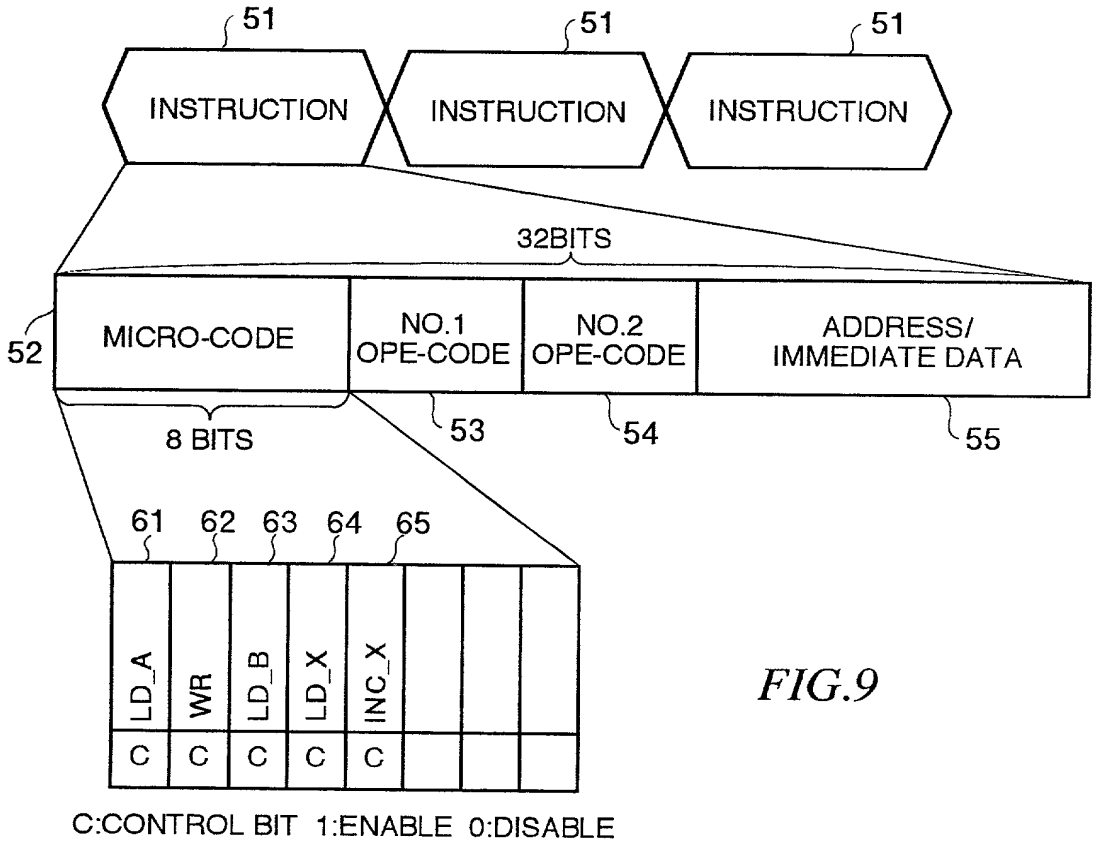
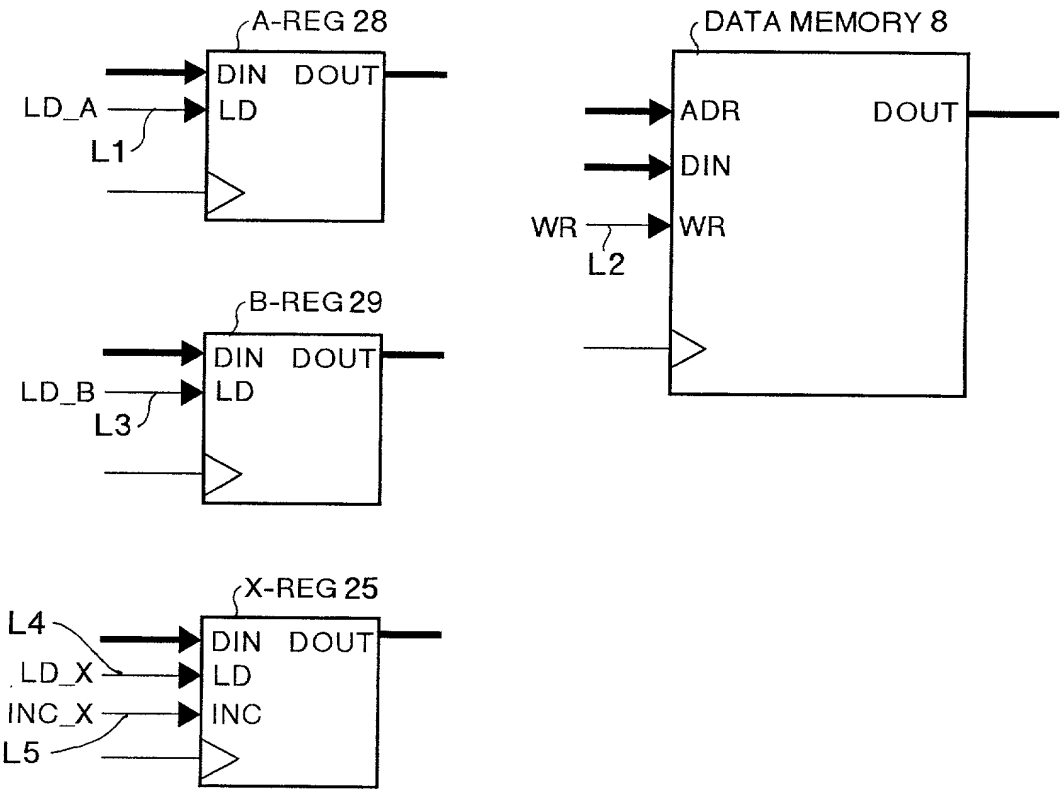

51
51
51
INSTRUCTION
INSTRUCTION
INSTRUCTION
32BITS
52
MICRO-CODE
NO.1 OPE-CODE
NO.2 OPE-CODE
ADDRESS/ IMMEDIATE DATA
8 BITS
53
54
55
61
62
63
64
65
LD_A
WR
LD_B
LD_X
INC_X
C
C
C
C
C
C:CONTROL BIT 1:ENABLE 0:DISABLE

FIG.9


A-REG 28
DIN DOUT
LD_A
LD
L1
B-REG 29
DIN DOUT
LD_B
LD
L3
X-REG 25
L4
DIN DOUT
LD_X
LD
INC_X
INC
L5
DATA MEMORY 8
ADR
DOUT
DIN
WR
WR
L2

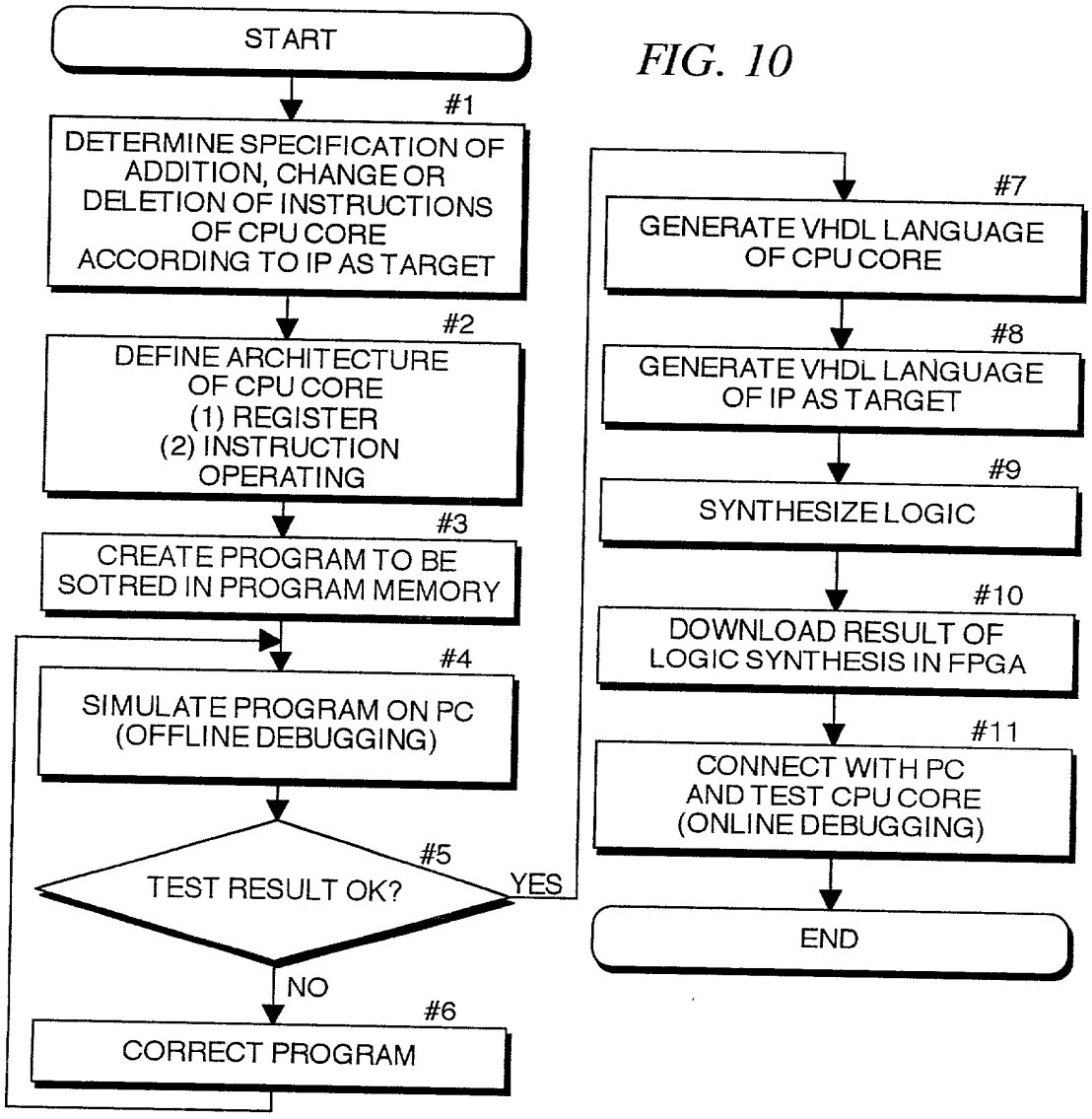

START
#1
DETERMINE SPECIFICATION OF ADDITION, CHANGE OR DELETION OF INSTRUCTIONS OF CPU CORE ACCORDING TO IP AS TARGET
#2
DEFINE ARCHITECTURE OF CPU CORE (1) REGISTER (2) INSTRUCTION OPERATING
#3
CREATE PROGRAM TO BE SOTRED IN PROGRAM MEMORY
#4
SIMULATE PROGRAM ON PC (OFFLINE DEBUGGING)
#5
TEST RESULT OK?
YES
NO
#6
CORRECT PROGRAM
#7
GENERATE VHDL LANGUAGE OF CPU CORE
#8
GENERATE VHDL LANGUAGE OF IP AS TARGET
#9
SYNTHESIZE LOGIC
#10
DOWNLOAD RESULT OF LOGIC SYNTHESIS IN FPGA
#11
CONNECT WITH PC AND TEST CPU CORE (ONLINE DEBUGGING)
END

FIG. 10

FIG.11

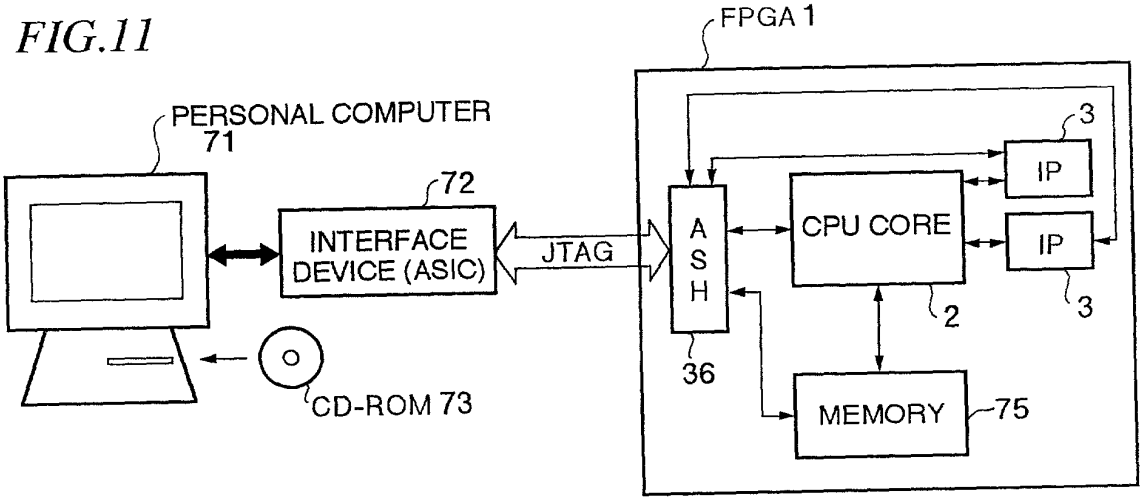
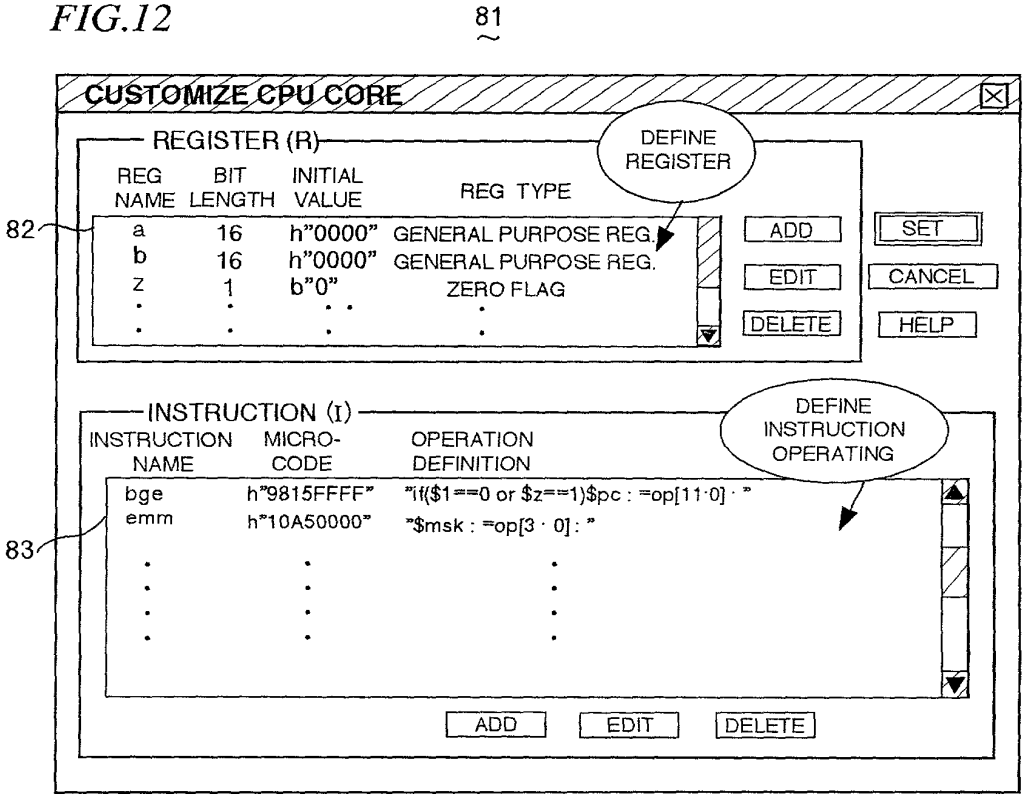

FPGA 1
PERSONAL COMPUTER
71
72
INTERFACE DEVICE (ASIC)
JTAG
A S H
36
CPU CORE
2
IP
3
IP
3
MEMORY
75
CD-ROM 73

FIG.12


81
CUSTOMIZE CPU CORE
REGISTER (R)
DEFINE REGISTER
REG NAME
BIT LENGTH
INITIAL VALUE
REG TYPE
82
a 16 h"0000" GENERAL PURPOSE REG.
b 16 h"0000" GENERAL PURPOSE REG.
z 1 b"0" ZERO FLAG
ADD
EDIT
DELETE
SET
CANCEL
HELP
INSTRUCTION (I)
DEFINE INSTRUCTION OPERATING
INSTRUCTION NAME
MICRO-CODE
OPERATION DEFINITION
83
bge h"9815FFFF" "if($1==0 or $z==1)$pc : =op[11·0] · "
emm h"10A50000" "$msk : =op[3 · 0] : "
ADD
EDIT
DELETE